An Improved General E-unification Method

By: Dougherty, Daniel J & **Johann, Patricia**

Abstract

A generalization of Paramodulation is defined and shown to lead to a complete E-unification method for arbitrary equational theories E. The method is defined in terms of transformations on systems, building upon and refining results of Gallier and Snyder.

Dougherty, Daniel J & **Johann, Patricia** (1992) "An Improved General E-unification Method" *Journal of Symbolic Computation* 14 (1992), pp. 303 - 320. Published October 1992. ISSN: 0747-7171 version of record available from www.sciencedirect.com

DANIEL J. DOUGHERTY AND PATRICIA JOHANN ‡

(Received August 19, 1991)

1. Introduction

Let E be a set of equations. An *E-unifier* of terms A and B is a substitution θ such that θA and θB are equal under E. This paper considers the problem of E-unification for arbitrary equational theories E, and presents an inference rule approximating Paramodulation and leading to a complete E-unification procedure which generalizes Narrowing. This sheds some light on the boundary between arbitrary E-unification situations and E-unification under canonical E.

We embrace the point of view that transformations on systems represent a "proof theory" for E-unification, and can provide a framework for the development of unification procedures and a setting for proving completeness results. The main argument in Section 3 leads to a short proof of the completeness of a transformation version of Narrowing when E has a canonical presentation and an outline of a proof of the completeness of an improvement, Basic Narrowing, due to Hullot (1980).

Our work is a refinement of the general E-unification method of Gallier and Snyder (1989), and the most convenient way to describe our main result is to compare our procedure with theirs, in the next few paragraphs.

Given equations E whose left-to-right orientation gives a canonical rewrite system, Narrowing is a complete method for generating E-unifiers for a pair $\langle A,\ B\rangle$ which proceeds as follows: select from E the left-hand side of an equation $L = R$ and from $\langle A,\ B\rangle$ a non-variable subterm, say, A/u, and treat the pair $\langle A/u,\ L\rangle$ as a syntactic unification problem. If A/u and L have a most general unifier σ, apply σ to A and B, perform the rewrite step using $\sigma L = \sigma R$. Continue, composing substitutions at each step, until a unifiable pair is found.

Gallier and Snyder's main result shows that for an arbitrary set of equations E, a complete set of E-unifiers can be found if this method is generalized by allowing L to

†

be either side of an equation, relaxing the required relationship between A/u and L (cf. Definition 2.7 and considering the pair $\langle A/u, L\rangle$ as an E-unification problem to be solved, that is, added to the system.

Now suppose we say that two terms *top-unify* if they have the same symbol at each u which is a non-variable occurrence in both terms. Our main result is that for any E, a complete set of E-unifiers can be generated by a procedure that requires, in the notation above, that A/u and L top-unify and that the corresponding Term Decompositions be done immediately.

The restriction to top-unifying pairs limits the non-determinism in the method considerably. Perhaps more significantly, the difference between E-unification for E with canonical presentations and for arbitrary E can now be attributed to the distinction between unification and top-unification, together with the fact that canonical presentations orient the equations.

E-unification was introduced by Plotkin (1972). The explicit use of transformations on systems as an approach to unification is due to Martelli and Montanari (1982), but as Gallier and Snyder point out, it was implicit in Herbrand's (1930) dissertation. Kirchner (1984, 1985, 1986) was apparently the first to investigate the use of transformations in E-unification.

1.1. NOTATION

Terms $(A, B, C, \ldots)$ are built from *variables* $(x, y, z, \ldots)$ using a set of (one-sorted) first-order function symbols. Subterms are referenced in the usual way by sequences of positive integers called *occurrences* $(u, v, w, \ldots)$; we write A/u for the subterm of A at occurrence u, and write $A[u \leftarrow X]$ for the result of replacing A/u by X.

Equations are unordered pairs of terms; we use $\overset{E}{\longleftrightarrow}$ to denote the one-step equational inference relation. We use "$\equiv$" to denote syntactic identity and "$=_E$" to denote the equivalence relation generated by $\overset{E}{\longleftrightarrow}$.

Substitutions $(\sigma,\ \delta,\ \theta,\ \ldots)$ are endomorphisms of the term algebra leaving all but finitely many variables fixed; application of a substitution to a term is denoted by juxtaposition, as is composition of substitutions (so, for example, $\eta\sigma x \equiv \eta(\sigma(x))$. When σ is a substitution, the *domain* $D\sigma$ of σ is the set of variables x such that $\sigma x \not\equiv x$; σ is *idempotent* if $\sigma\sigma \equiv \sigma$, or equivalently if the set $I\sigma$ of variables occurring among those σx with $x \in D\sigma$ is disjoint from $D\sigma$. Write $\sigma \equiv \tau\ [V]$ (respectively, $\sigma =_E \tau\ [V]$) to indicate that for all $x \in V$, $\sigma x \equiv \tau x$ (respectively, $\sigma x =_E \tau x$); write $\sigma \leq \tau\ [V]$ (respectively, $\sigma \leq_E \tau$) if there is a substitution η such that $\eta\sigma \equiv \tau\ [V]$ (respectively, $\eta\sigma =_E \tau\ [V]$).

A relation $\sim$ on terms is *stable* if $A \sim B$ implies that for all substitutions θ, $\theta A \sim \theta B$; say that $\sim$ is *monotone* if $A \sim B$ implies that for all terms T and occurrences u in T, $T[u \leftarrow A] \sim T[u \leftarrow B]$. If $\Longrightarrow$ is any relation, we write $\Longrightarrow\!\!\!\!\!\!*\ \ $ for its relexive transitive closure.

Definitions and notation not presented here should be found in (Dershowitz and Jounnaud, 1991).

We always assume that our sets of equations are consistent.

2. Systems and E-unification methods

Transformation-based unification methods attempt to reduce *systems* representing unification problems to *solved* systems, from which solutions may be extracted immediately.

The transformations we consider for E-unification can introduce variables from the equations in E, but in our answer substitutions we will typically be concerned only with variables occurring in the original problem. It is important that the new variables be distinct from variables in the original system, but this requires some attention since the transformations may also delete variables from a system. To see the problem, suppose a variable x from the original system were deleted but then introduced later. Confusion would arise if the eventual answer were to bind x.

Now, when a procedure is implemented, a mechanism is provided to generate new variables distinct from those occurring in the past. It is tempting, in a formal treatment of transformations underlying such procedures, to simply declare that variables introduced may not have occurred in "previous" steps. But this obscures the distinction between transformations (as reductions of a problem) and procedure-steps (which are situated in time). An appeal to the computational history of a system compromises the key principle underlying the investigation of sets of transformations — that they abstract the logical content of unification procedures from considerations of control and data structures. For example, the naive approach makes it impossible to argue by induction over computations as sequences of transformations; note that a tail of such a computation would not properly be a computation at all, since its choice of variables would be conditioned by properties of "earlier" systems, not even appearing in the subsequence.

We will therefore construe systems as being explicitly tagged with a set of variables. When a system is considered as input to an E-unification procedure, the associated set of variables will be the set of variables occurring among the terms of the system; when a transformation introduces new variables, those will be added to the associated set. However, variables will never be deleted from the associated set of a system, even if a deletion-transformation removes all their occurences from the terms in a system. This explicit perseverance of variables corresponds to a mechanism in an implementation for recording the set of variables occurring "in the past". In particular, an instance of a procedure may now be faithfully modeled as a sequence of transformations.

Although we do not treat many-sorted logic in this paper, we note in passing that a proper treatment of many-sorted equational logic similarly requires (for different reasons) an explicit indication of a set of "relevant" variables; see, for example, (Goguen and Meseguer 1981, 1985).

DEFINITION 2.1. A *pair* $\langle A, B\rangle$ is a two-element multiset of terms. A *system* is a finite set $\mathcal{S}$ of pairs together with a finite set $Vars(\mathcal{S})$ of variables, including at least the variables occurring among the terms of $\mathcal{S}$. We will usually not need to explicitly indicate the set $Vars(\mathcal{S})$, and may abuse notation and speak of "the system $\mathcal{S}$".

A pair $\langle x, A\rangle$ is *solved* in $\mathcal{S}$, and x is a *solved variable* of $\mathcal{S}$, if there are no occurrences of x in a term pair other than the one indicated. If each pair in $\mathcal{S}$ is solved then $\mathcal{S}$ is a *solved system* and determines an idempotent substitution in an obvious way, although a pair consisting of two distinct solved variables requires a choice as to which of them is to be in the domain of the substitution. We will assume that a uniform method exists for making such a choice, and so will refer to *the* substitution determined by a solved system.

An *E-unifier* of a system is a simultaneous E-unifier of the pairs in the system; we identify syntactic unification with unification under the empty set of equations (and in this case may simply speak of a *unifier* of a system).

A *most general* (syntactic) unifier σ of a system $\mathcal{S}$ is an idempotent unifier with domain included in $Vars(\mathcal{S})$ such that for every unifier θ of $\mathcal{S}$, $\sigma \leq \theta[Vars(\mathcal{S})]$.

As is customary, we write $\mathcal{S}, \langle A, B\rangle$ to abbreviate $\mathcal{S} \cup \{\langle A,\ B\rangle\}$. Since this is ambiguous as a decomposition of the system in question ($\mathcal{S}$ may or may not contain $\langle A, B\rangle$), we introduce the notation $\mathcal{S}; \langle A, B\rangle$ to refer to $\mathcal{S} \cup \{\langle A,\ B\rangle\}$ with the understanding that $\langle A, B\rangle$ is *not* a pair in $\mathcal{S}$.

If σ is an idempotent substitution, write $[\sigma]$ for any solved system by which it is determined.

Martelli and Montanari (1982) defined a set of transformations in order to study syntactic unification. The following variant of Martelli and Montanari's transformations is defined by Gallier and Snyder (1989).

DEFINITION 2.2. The set of transformations for *Syntactic Unification* consists of the following. We indicate only the effect on the pairs of a system; each transformation below is to induce no change in the associated set of variables of a system.

Trivial:

$$\mathcal{S};\ \langle A, A\rangle \Longrightarrow \mathcal{S}$$

Term Decomposition:

$$\mathcal{S};\ \langle f(A_1, ..., A_n),\ f(B_1, ..., B_n)\rangle \Longrightarrow \mathcal{S},\ \langle A_1, B_1\rangle, ..., \langle A_n, B_n\rangle$$

Variable Elimination:

$$\mathcal{S};\ \langle x, A\rangle \Longrightarrow \varphi\mathcal{S},\ \langle x, A\rangle, \quad \text{where } \varphi \text{ is the substitution } \{x \mapsto A\},$$

provided $\langle x, A\rangle$ is not solved and x does not occur in A.

Observe the use of ";" on the left-hand sides of transformations, so that the effect of the transformation is unambiguous, and the use of "," on the right-hand sides, to preclude repetition of identical pairs.

These transformations naturally define a non-deterministic procedure, which we denote *Syntactic Unification*, or simply $\mathcal{SU}$. Write $\mathcal{S} \overset{\mathcal{SU}}{\Longrightarrow} \mathcal{S}'$ if $\mathcal{S}'$ is obtained from $\mathcal{S}$ in one $\mathcal{SU}$ step.

THEOREM 2.3. *(Martelli and Montanari 1982) Every $\mathcal{SU}$ computation terminates. If $\mathcal{S}$ is unifiable then every $\mathcal{SU}$ computation on $\mathcal{S}$ terminates in a solved system determining a most general unifier for $\mathcal{S}$. If $\mathcal{S}$ is not unifiable then no $\mathcal{SU}$ computation on $\mathcal{S}$ terminates in a solved system.*

PROOF. Associate with each system the number of unsolved variable occurrences and then the sum of the depths of the terms; order these pairs lexicographically and observe that any transformation decreases the associated pair. This proves termination. The remaining assertions follow from the facts that each transformation preserves the set of unifiers of a system, that an irreducible system is unifiable iff it is solved and that if $[\sigma]$ is a solved system, then σ is a most general unifier of $[\sigma]$. □

For a non-empty set E of equations, a procedure based on $\mathcal{SU}$ certainly cannot yield all E-unifiers of arbitrary systems, so we seek to add transformations to the set $\mathcal{SU}$.

We cannot hope for a set of transformations for general E-unification which performs as well as those for syntactic unification. E-unification is undecidable even under stringent conditions on E, most general E-unifiers do not necessarily exist, and in fact Fages and Huet (1986) have shown that there are equational theories E and systems $\mathcal{S}$ which do not possess $\leq_E$-minimal unifiers. Consequently we say that an E-unification procedure is *complete* (for E) if for every system $\mathcal{S}$ and every substitution θ which E-unifies $\mathcal{S}$, there is a computation on $\mathcal{S}$ yielding an E-unifying substitution σ with $\sigma \leq_E \theta\ [Vars(\mathcal{S})]$.

Of course we will expect soundness: an E-unification procedure is *sound* (for E) if it never returns substitutions which are not E-unifiers. All of the procedures considered in this paper are restrictions of those considered by Gallier and Snyder. It follows that the soundness of these procedures is an immediate consequence of the soundness of Gallier and Snyder's method, and need not be discussed further.

The Paramodulation inference rule was introduced by Robinson and Wos (1969) in the context of first-order theorem proving in the presence of equality axioms. Narrowing was a refinement proposed by Slagle (1974) and by Lankford (1975) to take advantage of a canonical term rewriting presentation of the relevant equational theory; Fay (1979) investigated Narrowing as the basis of an E-unification procedure.

DEFINITION 2.4. Let $\langle A, B\rangle$ be a pair in $\mathcal{S}$, let u be a non-variable occurrence of A, and let $L = R$ be a variant of an equation in E whose variables do not occur in $\mathcal{S}$ such that σ is a most general unifier of A/u and L.

1 The following is an instance of *Paramodulation*:

$$\mathcal{S} \equiv \mathcal{S}';\ \langle A, B\rangle \quad \Longrightarrow \quad \sigma\mathcal{S}',\ [\sigma],\ \sigma\langle A[u \leftarrow R], B\rangle.$$

If V is the set of variables associated with the left-hand system and W is the set of variables occurring in the equation $L = R$, then $V \cup W$ is the set of variables associated with the right-hand system. We call $\langle A/u, L\rangle$ the *witness* pair.

2 If the equations in E are oriented from left to right, a *Narrowing* step is a Paramodulation step in which the witness pair uses the left-hand side of an equation.

3 *Narrowing* is the non-deterministic procedure determined by Narrowing steps and Syntactic Unification steps, with the further restriction that Syntactic Unification steps are performed only on witness pairs and during a final stage, computing a solved system from a unifiable one.

We emphasize that the definition of Paramodulation here forbids "paramodulation into variables", that is, the A/u subterm above may not be a variable.

The choice of witness pair represents a guess that an equational derivation between substitution instances of A and B has an initial step using equation $L = R$ at occurrence u.

THEOREM 2.5. *(Fay 1979)* Narrowing *is sound and complete for sets of equations whose left-to-right orientation induces a canonical rewrite system.*

We outline a proof at the end of Section 3.

Gallier and Snyder prove that *Narrowing* is complete under the weaker hypothesis that there exists a reduction ordering $\succ$ such that the $\succ$-oriented *ground* instances of the equations form a confluent term rewriting system.

The following example shows that even a procedure based on Paramodulation cannot be complete in general.

EXAMPLE 2.6. Let E consist of the two equations $f(a,b) = a$ and $a = b$, and let $\mathcal{S}$ be $\langle f(x,x),\ x\rangle$. $\mathcal{S}$ is E-unifiable and is not solved, but no Paramodulation or Syntactic Unification steps apply out of $\mathcal{S}$.

In order to accomodate general E-unification, Gallier and Snyder defined a generalization of Paramodulation.

DEFINITION 2.7. Let $\langle A, B\rangle$ be a pair in $\mathcal{S}$, let u be a non-variable occurrence of A, and let $L = R$ be a variant of an equation in E whose variables do not occur in $\mathcal{S}$ such that if L is not a variable then L and A/u have the same root function-symbol. Then the following is an instance of *Lazy Paramodulation*:

$$\mathcal{S} \equiv \mathcal{S}'; \langle A, B\rangle \quad \Longrightarrow \quad \mathcal{S}', \langle A/u, L\rangle, \langle A[u \leftarrow R], B\rangle,$$

with the additional requirement that when L is not a variable, Term Decomposition is immediately applied to the pair $\langle A/u, L\rangle$. If V is the set of variables associated with the left-hand system and W is the set of variables occurring in the equation $L = R$, then $V \cup W$ is the set of variables associated with the right-hand system.

Here, the choice of witness pair represents a guess that an equational derivation between substitution instances of A and B involves a step (not necessarily initial) using equation $L = R$ at occurrence u.

THEOREM 2.8. *(Gallier and Snyder 1989) The non-deterministic E-unification procedure determined by Lazy Paramodulation steps and Syntactic Unification steps is sound and complete for arbitrary sets E of equations. (Syntactic Unification steps are performed only on witness pairs and during a final unification stage.)*

The requirements that the terms in the witness pair have the same head symbol and that Term Decomposition be immediately applied may be seen as an attempt to retain some of the discipline of Paramodulation, specifically by restricting the number of candidate occurrences at which the transformation may be applied.

The completeness proof presented in (Gallier and Snyder 1989) overlooks the justification of the "same root symbol" constraint on the witness pair (see in particular the proof of Lemma 6.7 there). Their argument does, however, show completeness of a version of Lazy Paramodulation in which the relationship between terms in the witness pair is unconstrained. Furthermore, their intuition about the constraint was correct: the notion of *top-unification* we present below refines the one they impose, and our main theorem will show that we may in fact insist that A/u and L top-unify without sacrificing completeness.

DEFINITION 2.9. A and B *top-unify* if A and B have the same symbol at each u which is a non-variable occurrence in both terms.

The applications of top-unification in our setting will always concern variable-disjoint

terms A and B. In such a situation, top-unification coincides with unification when the terms in question are linear (i.e., have no repeated variables).

A useful characterization of top-unification is given in the next Lemma.

EMMA 2.10. *For any terms A and B, the following are equivalent.*

1. *A and B top-unify.*
2. *When Term Decomposition is applied as many times as possible starting with the pair $\langle A, B\rangle$, each pair in the resulting system has a variable as one of its elements.*

PROOF. Immediate from the definition. □

It will be convenient to write $dec\langle A, B\rangle$ for the system obtained by applying to $\langle A, B\rangle$ as many Term Decompositions as possible.

We can now give the refinement of Lazy Paramodulation which is the subject of this paper.

DEFINITION 2.11. Let $\langle A, B\rangle$ be a pair in $\mathcal{S}$, let u be a non-variable occurrence of A, and let $L = R$ be a variant of an equation in E whose variables do not occur in $\mathcal{S}$ such that A/u and L top-unify.

1. The following is an instance of *Relaxed Paramodulation*.
$$\mathcal{S} \equiv \mathcal{S}';\ \langle A, B\rangle \quad \Longrightarrow \quad \mathcal{S}',\ dec\langle A/u, L\rangle,\ \langle A[u \leftarrow R], B\rangle$$
If V is the set of variables associated with the left-hand system and W is the set of variables occurring in the equation $L = R$, then $V \cup W$ is the set of variables associated with the right-hand system.
2. The non-deterministic procedure determined by Relaxed Paramodulation steps and Syntactic Unification steps is denoted $\mathcal{RP}$. Write $\mathcal{S} \overset{\mathcal{RP}}{\Longrightarrow} \mathcal{S}'$ if $\mathcal{S}'$ is obtained from $\mathcal{S}$ in one $\mathcal{RP}$ step.

Our main result (Theorem 4.8) is that $\mathcal{RP}$ is complete for arbitrary E.

EXAMPLE 2.12. $\mathcal{RP}$ can simulate equational deduction. Specifically, suppose A occurs in a system and a subterm of A matches one side of an equation $L = R$, say $A/u \equiv \theta L$. Then $\mathcal{RP}$ can replace A/u by θR:
$$\begin{aligned} \mathcal{S};\ \langle A, B\rangle \quad &\overset{\mathcal{RP}}{\Longrightarrow} \quad \mathcal{S},\ dec\langle A/u, L\rangle,\ \langle A[u \leftarrow R], B\rangle \\ &\overset{\mathcal{SU}}{\Longrightarrow} \quad \mathcal{S},\ [\theta]\ \langle A[u \leftarrow \theta R], B\rangle \end{aligned}$$
where we have used the fact that $D\theta$ is disjoint from the variables of the original system. Observe that the solved subsystem corresponding to the matching substitution θ appears in the transformed system.

Later (in Lemma 4.5) we will see that Relaxed Paramodulations can simulate the construction of critical pairs.

EXAMPLE 2.13. Let E and $\mathcal{S}$ be as in Example 2.6. The E-unifier $\{x \mapsto a\}$ of $\mathcal{S}$ can be generated by $\mathcal{RP}$:

$$\begin{array}{lll} \langle f(x,x),x\rangle & \stackrel{\mathcal{RP}}{\Longrightarrow} & dec\langle f(x,x),f(a,b)\rangle,\ \langle a,x\rangle \\ & \stackrel{\mathcal{SU}}{\Longrightarrow} & \langle x,a\rangle,\langle a,b\rangle \\ & \stackrel{\mathcal{RP}}{\Longrightarrow} & \langle x,a\rangle, \end{array}$$

where the third line is derived by simulating the equational step replacing a by b and eliminating the resulting trivial pair.

EXAMPLE 2.14. $\mathcal{RP}$ can show the failure of E-unifiability of a system. Let E consist of the two equations $f(h(a)) = a$ and $a = b$, and let $\mathcal{S}$ be $\langle f(g(x)),\ x\rangle$. Then $\mathcal{S}$ is not solved, and no $\mathcal{RP}$ step applies out of $\mathcal{S}$. Anticipating the completeness theorem, we can conclude that $\mathcal{S}$ has no E-unifiers.

EXAMPLE 2.15. Let E be the usual presentation of group theory:

$$0 + z = z$$

$$(-y) + y = 0$$

$$(u + v) + w = u + (v + w).$$

We can verify the theorem $(-0) + x = x$ using $\mathcal{RP}$. In the sequence below, the first step uses the third group axiom, the second step uses the first axiom and the second pair of the system, the third step uses Variable Eliminations, and the final two $\mathcal{RP}$ steps perform equational steps as described in Example 2.12.

$$\begin{array}{lll} \langle (-0)+x,x\rangle & \stackrel{\mathcal{RP}}{\Longrightarrow} & \langle -0,u\rangle,\langle x,v+w\rangle,\langle (u+v)+w,x\rangle \\ & \stackrel{\mathcal{RP}}{\Longrightarrow} & \langle u,-0\rangle,\langle v,0\rangle,\langle w,z\rangle,\langle z,x\rangle,\langle (u+v)+w,x\rangle \\ & \stackrel{\mathcal{SU}}{\Longrightarrow} & \langle u,-0\rangle,\langle v,0\rangle,\langle w,x\rangle,\langle z,x\rangle,\langle ((-0)+0)+x,x\rangle \\ & \stackrel{\mathcal{RP}}{\Longrightarrow} & \langle u,-0\rangle,\langle v,0\rangle,\langle w,x\rangle,\langle z,x\rangle,\langle y,0\rangle,\langle 0+x,x\rangle \\ & \stackrel{\mathcal{RP}}{\Longrightarrow} & \langle u,-0\rangle,\langle v,0\rangle,\langle w,x\rangle,\langle z,x\rangle,\langle y,0\rangle,\langle z',x\rangle. \end{array}$$

The last system is solved, yielding a substitution σ with $x \notin D\sigma$. This gives the identity substitution as an E-unifier of the original system.

The next two sections prove the completeness of $\mathcal{RP}$.

3. Completeness in a special case

In this section we give a proof of the completeness of $\mathcal{RP}$ when the equational theory satisfies a certain closure property. In the next section we show how to lift this proof to obtain completeness for an arbitrary set of equations. This two-step strategy, of first assuming a kind of completeness for the underlying equations and then lifting that restriction, is the same as that used by Gallier and Snyder.

Gallier and Snyder observe that completeness of an E-unification procedure is implied by completeness with respect to *ground* substitutions; the justification involves replacing variables by Skolem-constants and showing that an answer substitution can be recovered

from its Skolemized version. Their first step, then, is to show completeness for ground substitutions when the equation-instances orientable with respect to a certain reduction order form a system which is ground confluent with respect to this ordering. The construction of these systems is essentially an *unfailing completion* procedure similar to those described in (Bachmair, Dershowitz, and Hsiang 1986), (Bachmair, Dershowitz, and Plaisted 1987), and (Bachmair 1987).

Gallier and Snyder's first step might be roughly summarized as: replace variables by new constants, work in the more congenial ground setting, then translate back to variables. It seems to us that the success of such a transfer to ground systems relies on the observation that although Narrowing requires a (canonical) rewrite relation capturing the given equational theory, the fact that the rewrite relation is preserved under substitution plays no role. This suggests eliminating the explicit passage to ground terms and simply treating variables as though they were constants. It seems worthwhile to pursue this more naive approach, if only as another point of view on the Skolemization trick. This is content of the current section.

The second step in Gallier and Snyder's proof is to show how to simulate a unification computation using the completed set of equations by a computation using the original set. As observed there, the former computation is essentially a *Narrowing* computation; in particular, the witness pairs at each step are syntactically unifiable. By doing a more delicate simulation — in the next section — we are able to retain part of that relationship, by arranging that the witness pairs top-unify.

Gallier and Snyder introduce a novel formalization of equational proofs, as certain sets of trees. We use ordinary equational derivations.

The following notion is an abstraction of the notion of canonical rewrite system.

DEFINITION 3.1. Fix a binary relation $>$ on terms.

For a set C of equations, let $>^C$ denote $(> \cap \overset{C}{\longleftrightarrow})$, and say that C is *closed with respect to* $>$ (or simply *closed*) if $>^C$ is monotone, noetherian, confluent, and has symmetric closure equal to $\overset{C}{\longleftrightarrow}$.

A term M is *minimal with respect to* $>$ (or simply *minimal*) if there does not exist an N such that $M >^C N$; a substitution θ is *minimal* if for all x in $D\theta$, θx is minimal.

Canonical rewrite systems provide the paradigm for closed sets: for any E, if the equations in E can be oriented so that the resulting rewrite system R is canonical, then E is easily seen to be closed by taking $>$ to be $\overset{R}{\longrightarrow}$, and the minimal terms are precisely the R-normal forms.

Similarly, for any E, if $>$ is a monotone noetherian relation whose symmetric closure contains $\overset{E}{\longleftrightarrow}$, then $>^E$ will inherit these properties (and its symmetric closure will equal $\overset{E}{\longleftrightarrow}$). It is easy to construct such relations $>$; the difficulty in building a closed set will be in enforcing confluence. We will see in the next section that whenever E contains all of its critical equations there exist relations with respect to which E is closed.

When C is closed the relation $>^C$ behaves in many ways like a canonical rewrite system, although it is not necessarily stable, and minimal terms correspond to normal forms. The next few paragraphs defend this analogy.

Since $>^C \subseteq \overset{C}{\longleftrightarrow}$, whenever $T >^C U$ then this fact is witnessed by a deduction step:

$$T \equiv T[u \leftarrow \delta L] \overset{C}{\longleftrightarrow} T[u \leftarrow \delta R] \equiv U$$

for some $L = R$ from C. Say that such a step is *minimal* if the substitution δ is minimal on the variables of L and R.

We also note that if $L = R$ is an equation in C and $\delta L >^C \delta R$, then L is not a variable. If it were, that variable would be a subterm of R (recall that we are assuming consistency of our equations), so that δL would be a proper subterm of δR. But in light of the monotonicity of $>^C$ this contradicts the fact that $>^C$ is noetherian.

It follows that variables are minimal.

Since $>^C$ and $\overset{C}{\longleftrightarrow}$ generate the same equivalence relation and $>^C$ is confluent, every term T is convertible under C with a *unique* minimal term.

We next define the objects corresponding to rewrite proofs in our setting.

DEFINITION 3.2. A $>^C$-proof between terms A and B is a pair of sequences of $>^C$-steps:

$$A \gg^C X \text{ and } B \gg^C X$$

for some term X. The *length* of such a proof is the sum of the number of $>^C$-steps in the two sequences.

Say that a $>^C$-proof is *minimal* if each of its steps is minimal.

We see that minimal proofs are analogous to innermost rewrite proofs. The key fact for us is that terms can be $>^C$-reduced to their minimal forms by minimal proofs.

LEMMA 3.3. *Let C be closed with respect to $>^C$. If $A =_C B$ then there is a minimal $>^C$-proof between A and B.*

PROOF. Since A and B have the same minimal form, the lemma will be established if we show that any term T admits a minimal $>^C$-proof between it and its minimal form. The proof of this fact is by noetherian induction over $>^C$; it suffices to show that if T is not minimal then there exists some minimal $>^C$-step out of T.

If T is not minimal, choose u so that T/u is not minimal but every proper subterm of T/u is minimal. It follows that there is an equation $L = R$ from C and a substitution δ such that T/u is of the form δL and $\delta L >^C \delta R$. L is not a variable, so for each $x \in Vars(L)$, δx is a proper subterm of T/u, hence minimal.

Now, in contrast to rewrite systems, even though $\delta L >^C \delta R$ there may be variables of R not occurring in L. In this case, define $\delta' x$ to be (i) δx, when x is a variable of L, and (ii) the minimal form of δx, when x is in $Vars(R) - Vars(L)$, (iii) x, when x does not occur in L or R. Then δ' is a minimal substitution, $\delta' L >^C \delta' R$, and

$$T \equiv T[u \leftarrow \delta' L] \overset{C}{\longleftrightarrow} T[u \leftarrow \delta' R]$$

is a minimal $>^C$-step. □

DEFINITION 3.4. Fix a relation $>$ and a set C of equations closed with respect to $>^C$.

If $A =_C B$, the *degree* of $\langle A, B \rangle$ with respect to $>$ (or simply the *degree* of $\langle A, B \rangle$) is the length of a shortest minimal $>^C$-proof between A and B. The *degree* of $\mathcal{S}$ is the sum of the degrees of the pairs in $\mathcal{S}$, provided these degrees are all defined. When θ is a

substitution it will be convenient to refer to the degree of the pair $\langle \theta A, \theta B \rangle$ (respectively, of the system $\theta\mathcal{S}$) as the "θ-degree" of $\langle A, B \rangle$ (respectively, of $\mathcal{S}$).

The next lemma corresponds to the lifting lemma used in the standard proof of the completeness of *Narrowing*.

EMMA 3.5. *Let C be closed. If θ is a minimal C-unifier of system $\mathcal{S}$ and the θ-degree of $\mathcal{S}$ is positive, then there is a C-unifier θ_1 of $\mathcal{S}$ and a Relaxed Paramodulation transformation $\mathcal{S} \stackrel{\mathcal{RP}}{\Longrightarrow} \mathcal{S}_1$ such that*

1 $\theta_1 \equiv \theta\ [Vars(\mathcal{S})]$,
2 θ_1 is minimal,
the witness pair for this step syntactically unifies, and
4 the θ_1-degree of $\mathcal{S}_1$ is less than the θ-degree of $\mathcal{S}$.

PROOF. We may assume without loss of generality that $D(\theta) \subseteq Vars(\mathcal{S})$. Choose $\langle A, B \rangle$ from $\mathcal{S}$ with positive θ-degree, and consider a shortest minimal $>^C$-proof between θA and θB. At least one of A and B is not a variable since θ is minimal; we may assume that there is a $>^C$-step out of θA. Write this step as

$$\theta A \equiv (\theta A)[u \leftarrow \delta L] >^C (\theta A)[u \leftarrow \delta R]$$

in which δ is minimal and, with a suitable choice of equation variant, $D\delta \cap Vars(\mathcal{S}) = \emptyset$. Since θ is minimal, u is a non-variable occurrence in A.

Take θ_1 to be $\theta \cup \delta$. The first assertion is clear. The substitution θ_1 is minimal by the minimality of δ and by hypothesis on θ.

$\mathcal{S}_1$ is determined by the following transformation:

$$\mathcal{S} \equiv \mathcal{S}'; \langle A, B \rangle \quad \stackrel{\mathcal{RP}}{\Longrightarrow} \quad \mathcal{S}', dec\langle A/u, L \rangle, \langle A[u \leftarrow R], B \rangle \equiv \mathcal{S}_1.$$

The third assertion of the lemma holds since θ_1 unifies A/u and L.

The final claim follows from the observations that the θ_1-degree of $\langle A/u, L \rangle$ is 0 (and hence so is the degree of $dec\langle A/u, L \rangle$), and the θ_1-degree of $\langle A[u \leftarrow R], B \rangle$ is less than the θ_1-degree of $\langle A, B \rangle$, while θ_1 agrees with θ on $\langle A, B \rangle$ and on $\mathcal{S}'$. □

Of course, the proof of Lemma 3.5 shows that *any* minimal $>^C$-step can be lifted to a Relaxed Paramodulation.

THEOREM 3.6. *$\mathcal{RP}$ is complete for closed C.*

PROOF. Let θ be a C-unifier of $\mathcal{S}$. We wish to show that there is a computation on $\mathcal{S}$ yielding a substitution σ with $\sigma \leq_C \theta\ [Vars(\mathcal{S})]$. We may assume without loss of generality that θ is minimal.

The proof is by induction on the θ-degree of $\mathcal{S}$.

If the θ-degree of $\mathcal{S}$ is 0 then θ is a unifier of $\mathcal{S}$, and a sequence of $\mathcal{SU}$ transformations can return a most general unifier σ.

Otherwise, by Lemma 3.5 there is a Relaxed Paramodulation transformation out of $\mathcal{S}$ yielding $\mathcal{S}_1$ and a minimal C-unifier θ_1 of $\mathcal{S}_1$ such that $\theta_1 \equiv \theta\ [Vars(\mathcal{S})]$ and such that the θ_1-degree of $\mathcal{S}_1$ is less than the θ-degree of $\mathcal{S}$. By induction, there is a $\mathcal{RP}$ computation out of $\mathcal{S}_1$ computing a C-unifier σ of $\mathcal{S}_1$ with $\sigma \leq_C \theta_1\ [Vars(\mathcal{S}_1)]$. By soundness of the

transformations, σ is a C-unifier of $\mathcal{S}$. Since $Vars(\mathcal{S}) \subseteq Vars(\mathcal{S}_1)$, $\sigma \leq_C \theta_1\ [Vars(\mathcal{S})]$. But since $\theta_1 \equiv \theta\ [Vars(\mathcal{S})]$, $\sigma \leq_C \theta\ [Vars(\mathcal{S})]$ as desired. □

We have pointed out that if the equations of some E can be oriented to form a canonical term rewriting system R, then E is *closed* with respect to the rewrite relation $\xrightarrow{R}$. The argument above almost provides a proof of the completeness of Narrowing in this situation, but not quite. The reason is that a Relaxed Paramodulation step is not a Narrowing step even when the witness pair unifies — in Narrowing one must actually unify the witness pair as part of the transformation, and this has the effect (via the associated Variable Eliminations) of applying the unifying substitution to the entire system. But it is easy to see that Lemma 3.5 holds with Narrowing steps in place of Relaxed Paramodulation steps: it suffices to observe that when $\mathcal{S}_1$ undergoes any $\mathcal{SU}$ step (during the process of computing the unifier of the witness pair) its θ_1-degree is unchanged. The only non-trivial case is Variable Elimination: suppose such a step uses $\langle x, A\rangle$ as a redex pair, and write φ for the substitution $\{x \mapsto A\}$. Note that the fact that $\theta_1 x \equiv \theta_1 A$ immediately implies that $\theta_1 \equiv \theta_1\varphi$, and therefore that the θ_1-degree of each pair will be undisturbed after application of φ. The argument in Theorem 3.6 then applies without change.

Hullot (1980) defined *Basic Narrowing* to embody the insight that one can forbid Narrowing at any of the occurrences created by the substitution being computed. The reader familiar with the terminology of that paper will find it easy to check that the sequence constructed in our completeness proof is automatically *based on* the set of non-variable occurrences in the original system. Thus the formalism here allows us to separately analyze applications of Paramodulation (in the theorem proper) and applications of partial answer substitutions (in the discussion of the previous paragraph), leading to a simple indication of the completeness of Basic Narrowing.

4. Completeness of Relaxed Paramodulation

The previous section showed how to compute E-unifiers using Relaxed Paramodulation relative to a closed set of equations. In this section we give a naive completion procedure to yield a closed set $C^*(E)$ from an arbitrary set E, and — almost — lift the original construction from $C^*(E)$ to E. The lifting is not perfect, since (i) we cannot orient the equations and (ii) unification of witness pairs will not be preserved by the simulation. Top-unification is preserved, however.

To obtain a closed set from an arbitrary set of equations, it will suffice to saturate the set with its critical equations.

Definition 4.1. For any set E of equations, a *critical equation* of E is an equation $\sigma R = \sigma L[v \leftarrow \sigma U]$, where $L = R$ and $T = U$ are equations in E, v is a nonvariable occurrence of L, and L/v and T are unifiable with most general unifier σ.

Let $C(E)$ be E together with its critical equations. Then define $C^0(E) = E$, $C^{k+1}(E) = C(C^k(E))$, and $C^*(E) = \bigcup C^k(E)$.

To motivate the next lemma, observe that it is easy to construct total orderings on terms which are monotone and noetherian. For example, define any total noetherian relation $>$ on the variables together with the function sysmbols, extend $>$ to terms by

comparing the size of terms, breaking ties by comparing head symbols, and, if necessary, comparing immediate subterms from left to right.

EMMA 4.2. *Let $>$ be any monotone noetherian total order on terms. Then for any E, $C^*(E)$ is closed with respect to $>^{C^*(E)}$.*

PROOF. Let us write C for $C^*(E)$.

It is clear that $>^C$, i.e., $(> \cap \overset{C}{\longleftrightarrow})$, is monotone and noetherian, and has symmetric closure equal to $\overset{C}{\longleftrightarrow}$. In order to show confluence of $>^C$, it suffices to show local confluence.

Suppose $M >^C N$ and $M >^C P$. If the redexes of the derivation steps witnessing the $>^C$-reductions are either disjoint or overlap below a variable occurrence of the larger, the monotonicity of $>$ ensures the existence of a common $>^C$-reduct. Otherwise, the facts that C contains all of its critical equations and that $>$ is total ensure that either $N >^C P$ or $P >^C N$. □

Now let E be any set of equations. If a system $\mathcal{S}$ is E-unifiable by a substitution θ, then by Theorem 3.6 and Lemma 4.2 there is a $\mathcal{RP}$ computation using equations from $C^*(E)$ yielding an E-unifier σ, with $\sigma \leq_{C^*(E)} \theta[Vars(\mathcal{S})]$. Since E and $C^*(E)$ are equivalent theories, $\sigma \leq_E \theta[Vars(\mathcal{S})]$. We need to show how to lift such a computation to a computation involving only E-equations. The key result is the Simulation Lemma below, which will imply that the critical equations added to build $C^*(E)$ can be removed in favor $\mathcal{RP}$ steps over E.

Some notation will be useful. If A and B top-unify, write $A \simeq B$. Write $O(A)$ for the set of occurrences in A, and use the convention that if v is not an occurrence in a term A, $A[v \leftarrow X] \equiv A$. When $v \in O(A)$, let $\perp_v (A)$ be the set of occurrences in A of minimal length among those incomparable with v (two sequences are incomparable if neither is a subsequence of the other).

The following easy lemma collects the facts about top-unification that we will need.

EMMA 4.3. *1 $\sigma A \simeq \delta B$ implies $A \simeq B$.*

2 When $v \in O(A) \cap O(B)$, $A \simeq B$ implies $A/v \simeq B/v$.
If $A \simeq B$ then $A[v \leftarrow X] \simeq B[v \leftarrow X]$ (whether v is an occurrence in these terms or not).

4 If $v \notin O(A)$ then $A \simeq \sigma(L[v \leftarrow U])$ implies $A \simeq L$.

5 Suppose $v \in O(A)$. If σ is a most general unifier of L/v and T, and $A \simeq \sigma(L[v \leftarrow U])$, then $A/v \simeq U$ and $A[v \leftarrow T] \simeq L$.

PROOF. The first three assertions are clear from Lemma 2.10. To prove (4), note that by (1), $A \simeq L[v \leftarrow U]$; then use (3).

The first part of (5) follows from (1) and (2). For the second part, note that $A[v \leftarrow T] \simeq \sigma(L[v \leftarrow T]) \equiv \sigma(L)[v \leftarrow \sigma T] \equiv \sigma L$; then apply (1). □

EMMA 4.4. *Suppose $A \simeq B$ and $v \in O(A) \cap O(B)$.*

1 $\perp_v (A) = \perp_v (B)$.

2 By a sequence of Term Decompositions,

$$\langle A,\ B\rangle \stackrel{\mathcal{SU}}{\Longrightarrow\!\!\!\!\!\!\rightarrow} \langle A/v,\ B/v\rangle, \{\langle A/w,\ B/w\rangle \mid w \in \perp_v (A)\}$$

PROOF. An easy induction on terms. □

In the remainder of the paper we will often have occasion to refer to systems of the form $\mathcal{S}, [\sigma]$, where $\mathcal{S}$ is known from the context. In such a situation we always intend that the associated set of variables of $\mathcal{S}, [\sigma]$ is the set $Vars(\mathcal{S}) \cup D\sigma \cup I\sigma$.

EMMA 4.5. (SIMULATION LEMMA) *Suppose*

$$\mathcal{S}_0 \stackrel{\mathcal{RP}}{\Longrightarrow} \mathcal{S}_1$$

relative to $C(E)$, and let W be a co-infinite set of variables. Then there exists a substitution σ with $D\sigma \cap W = \emptyset$ such that

$$\mathcal{S}_0 \stackrel{\mathcal{RP}}{\Longrightarrow\!\!\!\!\!\!\rightarrow} \mathcal{S}_1, [\sigma]$$

relative to E.

PROOF. If the given transformation is a standard unification transformation or a Relaxed Paramodulation relative to E itself there is nothing to prove (we may take σ to be the identity). So suppose $\mathcal{S}_0 \stackrel{\mathcal{RP}}{\Longrightarrow} \mathcal{S}_1$ by a Relaxed Paramodulation step involving the critical equation $\sigma R = \sigma(L[v \leftarrow U])$ from E, where σ is a most general unifier of L/v and T. Since σ is idempotent its domain is disjoint from the variables it introduces, and since W has infinite complement we may, without altering the critical equation, vary L and T so that $D\sigma \cap W = \emptyset$.

Having done so, we take this to be the desired σ.

In justifying that σ works, we observe that there are two possible forms for the transformation $\mathcal{S}_0 \stackrel{\mathcal{RP}}{\Longrightarrow} \mathcal{S}_1$:

$$\text{I. } \mathcal{S}_0 \equiv \mathcal{S}'; \langle A, B\rangle \stackrel{\mathcal{RP}}{\Longrightarrow} \mathcal{S}', dec\langle A/u, \sigma R\rangle, \langle A[u \leftarrow \sigma(L[v \leftarrow U])], B\rangle \equiv \mathcal{S}_1,$$

and

$$\text{II. } \mathcal{S}_0 \equiv \mathcal{S}'; \langle A, B\rangle \stackrel{\mathcal{RP}}{\Longrightarrow} \mathcal{S}', dec\langle A/u, \sigma(L[v \leftarrow U])\rangle, \langle A[u \leftarrow \sigma R], B\rangle \equiv \mathcal{S}_1.$$

Considering each case separately, we mimic the critical-equation-dependent derivations by the following derivations which use equations only from E. Each Relaxed Paramodulation step in the simulations will be justified by Lemma 4.3.

When the transformation is as in I,

$$\begin{array}{rcl}
\mathcal{S}_0 & \equiv & \mathcal{S}'; \langle A,\ B\rangle \\
& \stackrel{\mathcal{RP}}{\Longrightarrow} & \mathcal{S}', dec\langle A/u,\ R\rangle, \langle A[u \leftarrow L],\ B\rangle \\
& \stackrel{\mathcal{RP}}{\Longrightarrow} & \mathcal{S}', dec\langle A/u,\ R\rangle, dec\langle L/v,\ T\rangle, \langle A[u \leftarrow L[v \leftarrow U]],\ B\rangle \\
& \stackrel{\mathcal{SU}}{\Longrightarrow\!\!\!\!\!\!\rightarrow} & \mathcal{S}', \sigma(dec\langle A/u, R\rangle), [\sigma], \langle A[u \leftarrow \sigma(L[v \leftarrow U])],\ B\rangle \\
& \stackrel{\mathcal{SU}}{\Longrightarrow\!\!\!\!\!\!\rightarrow} & \mathcal{S}', dec\langle A/u,\ \sigma R\rangle, [\sigma], \langle A[u \leftarrow \sigma(L[v \leftarrow U])],\ B\rangle \\
& \equiv & \mathcal{S}_1, [\sigma].
\end{array}$$

Notice that in passing from the third to the fourth line above we use $\mathcal{SU}$ to compute σ. We also use the fact — extending the "*dec*" notation to systems in the obvious way — that $dec(\sigma\,\langle A,\ B\rangle\,) \equiv dec(\sigma(dec\,\langle A,\ B\rangle\,))$.

In case the transformation is as in II, we have two subcases, according to whether v is a non-variable occurrence in A/u or not. The difference lies in the form of the subsystem of $\mathcal{S}_1$ represented by $dec\langle A/u, \sigma(L[v \leftarrow U])\rangle$.

When v is a non-variable occurrence of A/u then $dec\langle A/u, \sigma(L[v \leftarrow U])\rangle$ is

$$dec\langle A/uv, \sigma U\rangle, \{dec\langle A/uw, \sigma L/w\rangle | w \in \perp_v (L)\}.$$

Then, using E,

$$\begin{array}{lcl}
\mathcal{S}_0 & \equiv & \mathcal{S}'; \langle A, B\rangle \\
& \stackrel{\mathcal{RP}}{\Longrightarrow} & \mathcal{S}', dec\langle A/uv, U\rangle, \langle A[uv \leftarrow T], B\rangle \\
& \stackrel{\mathcal{RP}}{\Longrightarrow} & \mathcal{S}', dec\langle A/uv, U\rangle, dec\langle A[uv \leftarrow T]/u, L\rangle, \langle (A[uv \leftarrow T])[u \leftarrow R], B\rangle \\
& \equiv & \mathcal{S}', dec\langle A/uv, U\rangle, dec\langle A/u[v \leftarrow T], L\rangle, \langle A[u \leftarrow R], B\rangle \\
& \equiv & \mathcal{S}', dec\langle A/uv, U\rangle, dec\langle T, L/v\rangle, \\
& & \{dec\,\langle A/u[v \leftarrow T]/w,\ L/w\rangle\ |\ w \in \perp_v (L)\}, \langle A[u \leftarrow R], B\rangle \\
& \stackrel{\mathcal{SU}}{\Longrightarrow} & \mathcal{S}', dec\langle A/uv, \sigma U\rangle, \langle A[u \leftarrow \sigma R], B\rangle, \\
& & \{dec\,\langle A/u[v \leftarrow \sigma T]/w,\ \sigma L/w\rangle\ |\ w \in \perp_v (L)\}, [\sigma], \\
& \equiv & \mathcal{S}', dec\langle A/uv, \sigma U\rangle, \langle A[u \leftarrow \sigma R], B\rangle, \\
& & \{dec\,\langle A/uv,\ \sigma L/w\rangle\ |\ w \in \perp_v (L)\}, [\sigma], \\
& \equiv & \mathcal{S}_1, [\sigma]
\end{array}$$

When v is not a non-variable occurrence in A/u, there is a prefix v' of v such that v' is a variable occurrence in A/u. Let t be such that $v = v't$. Then $dec\langle A/u, \sigma(L[v \leftarrow U])\rangle$ is

$$\langle A/uv', \sigma(L/v'[t \leftarrow U])\rangle, \{dec\langle A/uw, \sigma L/w\rangle | w \in \perp_{v'} (L)\},$$

so that

$$\begin{array}{lcl}
\mathcal{S}_1 & \equiv & \mathcal{S}'; \langle A[u \leftarrow \sigma R], B\rangle, \langle A/uv', \sigma(L/v'[t \leftarrow U])\rangle, \\
& & \{dec\langle A/uw, \sigma L/w\rangle | w \in \perp_{v'} (L)\}.
\end{array}$$

We can then write

$$\begin{array}{lcl}
\mathcal{S}_0 & \equiv & \mathcal{S}';\ \langle A,\ B\rangle \\
& \stackrel{\mathcal{RP}}{\Longrightarrow} & \mathcal{S}', dec\,\langle A/u,\ L\rangle\,, \langle A[u \leftarrow R],\ B\rangle \\
& \equiv & \mathcal{S}',\ \langle A/uv',\ L/v'\rangle\,, \{dec\,\langle A/uw,\ L/w\rangle\ |\ w \in \perp_{v'} (L)\}, \langle A[u \leftarrow R],\ B\rangle \\
& \stackrel{\mathcal{RP}}{\Longrightarrow} & \mathcal{S}', dec\,\langle L/v't,\ T\rangle\,, \langle (L/v')[t \leftarrow U],\ A/uv'\rangle \\
& & \{\ ec\,\langle A/uw,\ L/w\rangle\ |\ w \in \perp_{v'} (L)\}, \langle A[u \leftarrow R],\ B\rangle \\
& \stackrel{\mathcal{SU}}{\Longrightarrow} & \mathcal{S}', [\sigma],\ \langle \sigma(L/v'[t \leftarrow U]),\ A/uv'\rangle\,, \\
& & \{\,\langle A/uw,\ \sigma L/w\rangle\ |\ w \in \perp_{v'} (L)\}, \langle A[u \leftarrow \sigma R],\ B\rangle \\
& \equiv & \mathcal{S}_1, [\sigma].
\end{array}$$

□

With each $\mathcal{RP}$ computation relative to $C^*(E)$ we can associate a multiset of natural numbers, with an occurrence of k in the multiset whenever an equation from $C^k(E)$ is used in a Relaxed Paramodulation step. The order on the natural numbers induces a well-founded order on this multiset, and our completeness theorem will induct over this order.

The Simulation Lemma introduces a residual solved system corresponding to the substitution involved in a critical pair construction – the next two lemmas verify that this causes no difficulties in the simulated computation.

Say that a substitution is *disjoint from* a system $\mathcal{S}$ if the domain of the substitution is disjoint from $Vars(\mathcal{S})$.

EMMA 4.6. *Suppose*

$$\mathcal{S} \stackrel{\mathcal{RP}}{\Longrightarrow} \mathcal{T}$$

and let σ be disjoint from $\mathcal{T}$. Then there is a substitution σ' with $D\sigma' \equiv D\sigma$ and

$$\mathcal{S}, [\sigma] \stackrel{\mathcal{RP}}{\Longrightarrow} \mathcal{T}, [\sigma'].$$

Furthermore, the multisets associated with these two computations are the same.

PROOF. Observe that in fact σ is disjoint from each system occurring in this computation, so it suffices to consider the case in which $\mathcal{S} \stackrel{\mathcal{RP}}{\Longrightarrow} \mathcal{T}$ consists of a single step.

If $\mathcal{T}$ is obtained by Trivial, Term Decomposition, or Relaxed Paramodulation we can apply the same transformation to $\mathcal{S}, [\sigma]$ and take σ' to be σ.

Suppose $\mathcal{S} \equiv \mathcal{S}'; \langle x, A\rangle$ and that $\mathcal{T}$ is obtained by Variable Elimination on x out of the pair $\langle x, A\rangle$; write φ for the substitution $\{x \mapsto A\}$. Write σ as $\{y_i \mapsto M_i | i \in I\}$. Then the following is an instance of Variable Elimination:

$$\begin{array}{rcl} \mathcal{S},\ [\sigma] & \equiv & \mathcal{S}',\ \langle x, A\rangle, \{y_i \mapsto M_i | i \in I\} \\ & \stackrel{\mathcal{SU}}{\Longrightarrow} & \varphi\mathcal{S}',\ \langle x, A\rangle, \{y_i \mapsto \varphi M_i | i \in I\}. \end{array}$$

Since $D\sigma \cap Vars(\mathcal{S}) = \emptyset$, each of the y_i is different from x and does not occur in A, so we may take σ' to be the substitution $\{y_i \mapsto \varphi M_i | i \in I\}$. □

EMMA 4.7. *Suppose*

$$\mathcal{S} \stackrel{\mathcal{RP}}{\Longrightarrow} [\delta]$$

and let σ be disjoint from $[\delta]$. Then for some substitution δ' such that $\delta' \equiv \delta\ [Vars([\delta])]$,

$$\mathcal{S}, [\sigma] \stackrel{\mathcal{RP}}{\Longrightarrow} [\delta'].$$

Furthermore, the multisets associated with the two computations are the same.

PROOF. The previous lemma yields a sequence

$$\mathcal{S}, [\sigma] \stackrel{\mathcal{RP}}{\Longrightarrow} [\delta], [\sigma'],$$

with the same multiset and with $D\sigma' \equiv D\sigma$.

Since $D\sigma' \cap D\delta = \emptyset$, applying Variable Elimination out of each of those pairs in $[\delta]$ which are not solved in $[\delta], [\sigma']$ yields the system determined by the pairs in $[\delta]$

together with each of the pairs $\langle x, \delta(\sigma' x)\rangle$ for x in $D\sigma'$. This latter system is solved since $D\sigma' \cap Vars([\delta]) = \emptyset$, and we may take it to be $[\delta']$.

The last assertion of the lemma holds since standard unification transformations do not contribute to the multiset associated with a computation. □

THEOREM 4.8. (COMPLETENESS THEOREM) *$\mathcal{RP}$ is a complete E-unification method for arbitrary sets of equations E.*

PROOF. Let θ be an E-unifier of $\mathcal{S}$. By Theorem 3.6 there is an $\mathcal{RP}$ computation using equations from $C^*(E)$ yielding a δ with $\delta \leq_E \theta\ [Vars(\mathcal{S})]$. It suffices, therefore, to show the following: *Whenever* $\mathcal{S} \overset{\mathcal{RP}}{\Longrightarrow} [\delta]$ *using equations in* $C^*(E)$ *then for some* δ^* *with* $\delta^* \equiv \delta\ [Vars(\mathcal{S})]$, $\mathcal{S} \overset{\mathcal{RP}}{\Longrightarrow} [\delta^*]$ *using equations in* E. We prove this by induction over the multiset associated with the sequence from $\mathcal{S}$ to $[\delta]$. In the base case, when there are no strictly positive members of the multiset, there is nothing to prove.

Otherwise we have

$$\mathcal{S} \overset{\mathcal{RP}}{\Longrightarrow} \mathcal{S}_0 \overset{\mathcal{RP}}{\Longrightarrow} \mathcal{S}_1 \overset{\mathcal{RP}}{\Longrightarrow} [\delta]$$

with the step from $\mathcal{S}_0$ to $\mathcal{S}_1$ using an equation from $C^k(E)$, $k > 0$. By the Simulation Lemma,

$$\mathcal{S}_0 \overset{\mathcal{RP}}{\Longrightarrow} \mathcal{S}_1, [\sigma]$$

using equations from $C^{k-1}(E)$, for some σ which is disjoint from $[\delta]$.

Applying the previous lemma to the sequence from $\mathcal{S}_1$ to $[\delta]$, we obtain

$$\mathcal{S} \overset{\mathcal{RP}}{\Longrightarrow} \mathcal{S}_0 \overset{\mathcal{RP}}{\Longrightarrow} \mathcal{S}_1, [\sigma] \overset{\mathcal{RP}}{\Longrightarrow} [\delta']$$

with $\delta' \equiv \delta\ [Vars([\delta])]$. Since $Vars(\mathcal{S}) \subseteq Vars([\delta])$ we have $\delta' \equiv \delta\ [Vars(\mathcal{S})]$. Furthermore, this latter computation submits to the induction hypothesis, since the Simulation Lemma traded the use of equations from $C^k(E)$ for equations from $C^{k-1}(E)$, and the previous lemma left the associated multiset unchanged.

Therefore

$$\mathcal{S} \overset{\mathcal{RP}}{\Longrightarrow} [\delta^*]$$

using equations in E, for some δ^* with $\delta^* \equiv \delta'\ [Vars(\mathcal{S})]$. But then $\delta^* \equiv \delta\ [Vars(\mathcal{S})]$ and the proof is complete. □

The question naturally arises whether $\mathcal{SU}$ steps can be safely applied at any time during an arbitrary $\mathcal{RP}$ computation. It is not hard to see that the answer is, in general, no, since application of Term Decomposition can prevent the lifting of root-rewrite steps. (For example, consider unifying $x + a$ and $y + b$ when $+$ is assumed commutative.)

Less straightforward is the issue of applying (via Variable Elimination) the partial substitutions which arise during a computation. This is the *Eager Variable Elimination* problem. The advantage of applying a partial answer substitution during the computation of an entire answer (as is done in Narrowing) is that filling out the term pairs to be E-unified can block some future Relaxed Paramodulation guesses by preventing the potential witness pair from top-unifying – this prunes the search space (especially in light of the discipline imposed by Basic Narrowing).

It is not hard to see that Variable Elimination does *not* change the set of E-unifiers of a system, but the strategy of Eager Variable Elimination for the transformations

appropriate to arbitrary sets of equations is not known to be complete (for a discussion in the context of Lazy Paramodulation, see (Gallier and Snyder 1989)). The technical problem is that, in contrast to Narrowing, the witness pairs of a Relaxed Paramodulation step need not have θ-degree equal to 0 (here, θ is the E-unifying substitution which the computation is trying to constuct). This forestalls a naive argument such as that given at the end of Section 3.

4.1. ACKNOWLEDGEMENT

The authors are indebted to Wayne Snyder for his enthusiasm and encouragement, and specifically for several instructive discussions.

References

BACHMAIR, L. (1987), *Proof Methods for Equational Theories*, dissertation, U. of Illinois, Urbana-Champaign.

BACHMAIR, L., DERSHOWITZ, N., AND HSIANG, J. (1986), Orderings for equational proofs, *Proc. Symp. on Logic in Computer Science*, 346–357.

BACHMAIR, L., DERSHOWITZ, N., AND PLAISTED, D. (1987), Completion without failure, *Proceedings of CREAS*.

DERSHOWITZ, N., AND JOUNNAUD, J.-P., (1991) Term Rewriting Systems, in *Handbook of Theoretical Computer Science*, 243–320, North-Holland, Amsterdam.

DOUGHERTY, D., AND JOHANN, P. (1990), An improved general E-unification method, *Proc. Tenth International Conference on Automated Deduction*, Lecture Notes in Artificial Intelligence **449** (ed. M. E. Stickel), 261–275, Springer-Verlag, New York.

FAY, M. (1979), First-order unification in an equational theory, *Proc. Fourth Workshop on Automated Deduction*.

FAGES, F., AND HUET, G. (1986), Complete sets of unifiers and matchers in equational theories, *Theoretical Computer Science* **43**, 189–200.

GALLIER, J. H., AND SNYDER, W. (1989), Complete sets of transformations for general E-unification, *Theoretical Computer Science* **67**, 203–260.

GOGUEN, J. A., AND MESEGUER, J. (1981), Completeness of many-sorted equational logic, *ACM SIGPLAN Notices*.

GOGUEN, J. A., AND MESEGUER, J. (1985), Completeness of many-sorted equational logic, *Houston Journal of Mathematics*, 307–334.

HERBRAND, J. (1930), *Sur la Theorie de la Demonstration*, dissertation; in *Logical Writings* (ed. W. Goldfarb), Cambridge, 1971.

HULLOT, J.-M. (1980), Canonical forms and unification, *Proc. Fifth International Conference on Automated Deduction*, Lecture Notes in Computer Science **87**, 318–334, Springer-Verlag, New York.

KIRCHNER, C. (1984), A new equational unification method: a generalization of Martelli-Montanari's algorithm, *Proc. Seventh International Conference on Automated Deduction*.

KIRCHNER, C. (1985), *Méthodes et Outils de Conception Systematique d'Algorithmes d'Unification dans les Theories Equationnelles*, Thèse d'Etat, Université de Nancy I.

KIRCHNER, C. (1986), Computing unification algorithms, *Proc. Symp. on Logic in Computer Science*, 206–216.

LANKFORD, D. (1975), *Canonical Inference*, Tech. Rep. # ATP-32, Dept. of Mathematics and Computer Science, U. Texas at Austin.

MARTELLI, A. AND MONTANARI, U. (1982), An efficient unification algorithm, *ACM Transactions on Programming Languages and Systems* **4**, 258–282.

MARTELLI, A., MOISO, C., AND ROSSI, G. F. (1986), An algorithm for unification in equational theories, *Proc. Third Conference on Logic Programming*.

PLOTKIN, G. (1972), Building in equational theories, *Machine Intelligence* **7** (ed. B. Meltzer and R. Michie), 73–90, Edinburgh University Press, Edinburgh.

ROBINSON G., AND WOS, L. (1969), Paramodulation and theorem-proving in first order theories with equality, *Machine Intelligence* **4** (ed. B. Meltzer and R. Michie), 135–150, Edinburgh University Press, Edinburgh.

SLAGLE J. R. (1974), Automated theorem proving for theories with simplifers, commutativity, and associativity, *Journal of the ACM* **12**, 23–41.